View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by RIT Scholar Works

Rochester Institute of Technology
RIT Scholar Works

Theses | Thesis/Dissertation Collections

8-2015

Pulsar: Design and Simulation Methodology for Dynamic Bandwidth Allocation in Photonic Network-on-Chip Architectures in Heterogeneous Multicore Systems

Kwadwo Opong-Mensah
kxo7118@rit.edu

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation
Opong-Mensah, Kwadwo, "Pulsar: Design and Simulation Methodology for Dynamic Bandwidth Allocation in Photonic Network-on-Chip Architectures in Heterogeneous Multicore Systems" (2015). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the Thesis/Dissertation Collections at RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.

Pulsar: Design and Simulation Methodology for Dynamic Bandwidth Allocation in Photonic Network-on-Chip Architectures in Heterogeneous Multicore Systems

by

Kwadwo Opong-Mensah

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering

Supervised by

Dr. Amlan Ganguly
Department of Computer Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, NY
August 2015

Approved By:

Dr. Amlan Ganguly
Primary Advisor – R.I.T. Dept. of Computer Engineering

Dr. Jing Zhang
Secondary Advisor – R.I.T. Dept. of Electrical Engineering

Dr. Andres Kwasinski
Secondary Advisor – R.I.T. Dept. of Computer Engineering

To those who have paved the way and to those who will follow in the coming generations.

Abstract

As the computing industry moved toward faster and more energy-efficient solutions, multicore computers proved to be dependable. Soon after, the Network-on-Chip (NoC) paradigm made headway as an effective method of connecting multiple cores on a single chip. These on-chip networks have been used to relay communication between homogeneous and heterogeneous sets of cores and core clusters. However, the variation in bandwidth requirements of heterogeneous systems is often neglected. Therefore, at a given moment, bandwidth may be in excess at one node while it is insufficient at another leading to lower performance and higher energy costs. This work proposes and examines dynamic schemes for the allocation of photonic channels in a Photonic Network-on-Chip (PNoC) as an alternative to their static-provision counterparts and proposes a method of simulating and selecting the characteristics of a dynamic system at the time of design as to achieve maximum system performance in a Photonic Network-on-Chip for a given application type.

Table of Contents

List of Figures

Chapter 1 Introduction

Fulfilling the demands of the computing industry to have higher component densities, lower power budgets and higher throughput capabilities requires that computer architectures must be transformed beyond the confines of conventional methodologies [2]. The advent of multicore computing allowed for multiple processors to share a single workload. However, conventional interconnection strategies applied to this new paradigm threatened to nullify the performance gains of multicore computing due to their high interconnection delay and energy costs [3]. While in conventional computer architecture designs, multicore or otherwise, the placement and routing of links between processing elements ("cores") was formerly a highly-neglected afterthought, the advent of the Network-on-Chip (NoC) brought into focus the necessity of optimization of not only the core architecture but the interconnection system as well.

The NoC paradigm allowed for decoupling of the processing element design from inter-core communication network design while still maintaining efficiency in both arenas. Various NoC-based solutions have been proposed. Some introduced Three-Dimensional VLSI (3DVLSI) networks a potential solution through which designers have three degrees of spatial freedom and therefore smaller network diameters [4]. Others turned to wireless transceivers as the backbone of the interconnection network between cores [5] [6]. Many have looked at the problems of area, power dissipation, throughput and latency of chip multi-processors (CMPs) through the lens of photonics [2] [7]. Consequently, many NoC architectures have been developed with photonic backbones. Moreover, 3D VLSI has been used in combination with photonics in many designs to keep the area overhead of the network to a minimum [7] [17].

The vast majority of Photonic Network-on-Chip (PNoC) architectures rely upon the principle of wavelength-division multiplexing (WDM). This principle manipulates light of various wavelengths to transmit information through a photonic interconnection network. Moreover, many of these architectures often use micro-ring resonators (MRRs) to implement WDM communication schemes [7] [8]. Wavelength-selective micro-ring resonators are extremely attractive for photonic network designers because of their small form factor. Equipped with various other devices as well, PNoCs use waveguides to carry information at the speed of light, making them a promising solution for multicore interconnection needs. However, depending of the type of traffic injected into the network, a given system can be overstressed, underutilized or both in certain sections.

Dynamic bandwidth allocation allows for a multicore system to adapt to the bandwidth needs of a particular task at any given moment. This work proposes and examines:

- A methodology for simulating dynamic bandwidth NoC traffic based on the mean and variance of the offered bandwidth demands along with the corresponding algorithms for generating demands
- A methodology for determining the optimal resource characteristics of a Photonic Network-on-Chip as to deliver higher performance than static-allocation alternatives for a given traffic type

In this work, a "resource" most commonly refers to a channel in Wavelength Division Multiplexing (WDM), but the concepts put forth can easily consider a resource to be a CDMA channel, a TDMA slot or some other network resource.

Chapter 2 Background

2.1. Integrated Optical Networks

In recent decades, progress in on-chip integration in the fields of photonics and plasmonics has enabled the development of more compact, more efficient devices and interconnects [9][10]. Advancements in waveguide integration as well as in the fabrication of resonator-based and electro-absorptive modulators [11] [12] [13] have fostered the development of on-chip interconnection networks which carry information between agents at extremely high data rates [14] [7] [8].

2.1.1 Sources

Just as electronic circuits require a supply of electrical power to operate, photonic circuits require a supply of optical power. This power typically comes in the form of an on-chip or off-chip laser [15] which injects photons into the network.

2.1.2 Couplers

Couplers are the on-ramps for light entering the highway of a photonic system. While electro-optic modulators can modulate various wavelengths of light once they are already present on the system's waveguides, couplers ensure that these wavelengths, upon ejection from a source, enter into the system, within the reach of modulators. Recent progress in the manufacture of on-chip couplers has accelerated the creation of on-chip networks [16].

2.1.3 Waveguides

Waveguides are the highways by which light travels in a photonic system. Once in the system, light propagates along waveguides. The PNoC layouts often run waveguides past the modulators first and then by the detectors to give all modulators a chance to modulate the various wavelengths before the detectors attempt to read data. Current research is moving towards lowering waveguide loss to ensure optimal signal detection regardless of propagation distance.

2.1.4 Modulators

Modulators act as on-off switches for light. Modulators come in many forms, but for the scope of this work, micro-ring resonators (MRRs), due to their pervasive presence in the NoC field, will be the modulator of choice. Micro-ring resonators, due to the principles of coupling and resonance, can block or steer the propagation of light under the right conditions [13]. When a waveguide is fabricated in the shape of a ring and is placed within a certain distance of another waveguide, it can couple light of a specific wavelength into or from itself. Due to electro-optic effects, electrical signals can be applied across the ring to vary its effective refractive index and consequently determine whether the ring resonates at a particular wavelength or not. Due to the sensitivity of these devices to temperature and process variations, they are often fabricated with an attached heating element which takes advantage of thermo-optic effects to keep themselves tuned to a desired frequency [17] [18] [19]. Due to the prevalence of silicon processes in the semiconductor industry, the field of silicon photonics has given birth to many modulator designs which feature micro-ring resonators. More recently, III-V semiconductors have also been used in developing next-generation photonic devices.

2.1.5 Routers

MRRs, due to their steering and blocking abilities, have been used to construct routers and filters which can use specific wavelengths to carry information between network nodes [20] [21]. Five port routers have been developed at about the scale of 0.3 mm^2 [21] which is about a tenth of the size of an AMD Jaguar core (3.1 mm^2). 3DVLSI has further reduced the impact of optical device area on photonic network implementation [7].

2.1.6 Detectors

Detectors are functionally the off-ramps by which light exits the photonic highway system. Following ejection from a source and modulation by one or more modulators, patterned data is picked up by detectors which convert the incident photonic signals back into electrical signals for processing by a destination agent. The same microring resonators which enable electro-optic modulation have been produced with sections of Germanium which to assist in the absorption of light and its conversion into electricity.

2.1.7 Other Components

The preceding components deal with the photonic portion of communication. However, prior to and following the transmission of data as optical signals, electrical signal processing takes place. High-speed transceivers and low-noise transimpedance amplifiers play a crucial role in the successful communication of data between agents [22].

2.2. Heterogeneous Processors

When dealing with a task that has many interdependencies, in most cases, a CPU is best suited for the job. In the case of parallel execution, GPUs often are the first choice of system architects. In the world of parallel computing, many systems have used CPUs and GPUs together. Parts of a task can be allotted to a CPU when they have many interdependencies and must be executed serially and, likewise, the parallel parts of a task can be handed off to a GPU for optimal performance, thereby getting the best of both worlds. However, the latencies and energy consumption associated with communication of data between discrete CPU and GPU chips make collaboration between the two a performance nightmare [28]. Recently, the fruits of work in the space of heterogeneous computing have begun to drive integration of CPU and GPU cores onto a single chip [23] [24] [25] [26] [37]. The replacement of inter-chip links with an on-chip interconnection network greatly reduces the latency and energy associated with communication between CPU and GPU cores [3].

2.3. Dynamic PNoCs

The introduction of more and more heterogeneous processors in rapid succession [27] points to the fact that computing tasks themselves are becoming more and more heterogeneous. Moreover, the bandwidth requirements of tasks can vary depending on their respective application-level needs or priority, e.g. the case of a real-time display application demanding a higher priority than that of a back-end computation.

In communication, the slowest element, i.e. the bottleneck of the system, typically determines the maximum bandwidth which a given connection can provide [28]. On-chip communication is no exception. For example, if the link between two agents is able to

provide a bandwidth 100 Terabits per second (Tbps) however the agents themselves are only capable of communicating at 100 Megabits per second (Mbps), the performance increase due to the link will be negligible. Meanwhile, the 100 Tbps resources allocated to these agents cannot be used by another on-going communication.

2.3.1 PNoC Topologies and Scaling

Photonic Network-on-Chip architectures exist in many forms. The crossbar network topology is commonly used due to its simplicity. Many works have developed hierarchical networks and other adaptations to improve scalability and performance with localized traffic patterns and performance overall [8] [30] [29], however, the crossbar has remained a fundamental building block of photonic networks. For this reason, the majority of this work focuses on the crossbar topology, but the principles are applicable to numerous others.

In the realm of photonic crossbars, exist the microcosms of Multiple-Write Single-Read (MWSR), Single-Write Multiple-Read (SWMR) and Multiple-Write Multiple-Read (MWMR) topologies [7] [8] [31][32]. These descriptions refer to the way in which nodes are connected to each other as shown in Figure 1. However, most of the previous work with these topologies in the Photonic Network-on-Chip field deal with Boolean allocation of resources (i.e. a network either has a full set of resources or none at all at a given moment). This work examines the effectiveness of allocation of resource amounts based on what a node needs at a given moment.

2.4. Evaluation of NoC Architectures

The fabrication of hardware, licensing of IP and other manufacturing considerations pertaining to on-chip networks can cost upwards of millions of dollars. Therefore, most research in the field of NoCs is conducted by way of simulators which consider empirical data from fabricated devices [33] [34] [35] [41] [42]. Experiments carried out in this work assume proper provisions at the physical layer, i.e. a laser power input into the system which is high enough to meet the threshold requirements of every detector in the system, a percentage of fabrication defects which is low enough as to not affect network component functionality, etc.

Chapter 3 System Architecture

3.1. Topology

The crossbar, having proved itself to be a fundamental building block of many photonic networks, is considered in the following evaluations, but the principles applied can be applied to hierarchical crossbars and other topologies.

In order for a system to support dynamic bandwidth allocation, certain provisions must be made in the hardware architecture. In many photonic systems, a Multiple-Write Single-Read (MWSR) architecture (Figure 1(a)) is used [7], where N is the total number of nodes in the photonic network and Node x is a node under observation. In this scheme, a given node is able to transmit on multiple channels but only listen on a single channel. In others, a Single-Write Multiple-Read (SWMR) scheme is employed as shown in Figure 1(b) [8]. In this scheme, a given node is able to transmit on a single channel and listen on multiple channels. However, in order to have a fully dynamic system, any node should be able to write to or read from any channel at the drop of a hat. This necessitates a Multiple-Write Multiple-Read (MWMR) architecture in the physical layer, shown in Figure 1(c), though a node only writes to a one channel at any given moment.

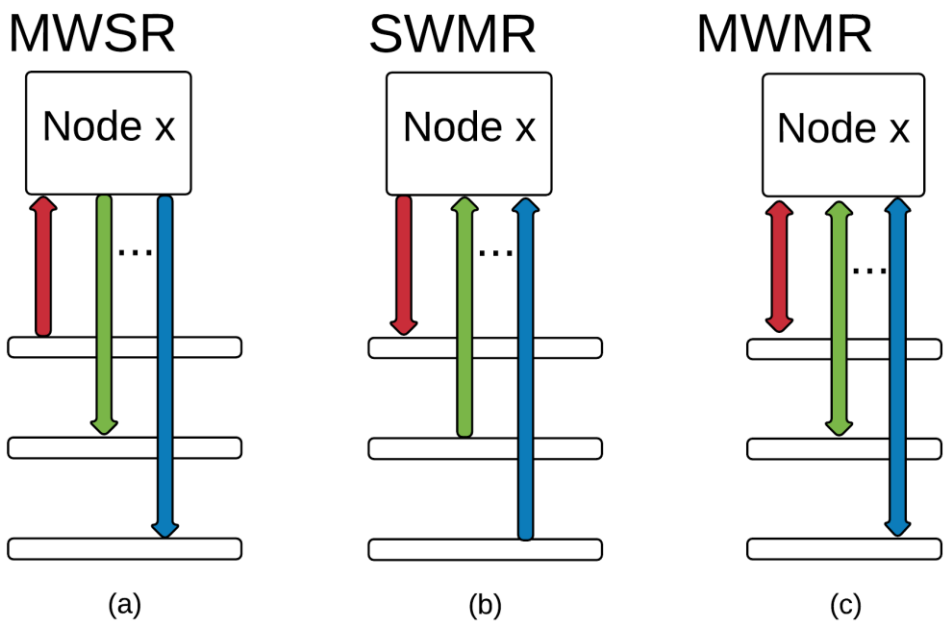

Figure 1: Node communication with waveguides for (a) MWSR, (b) SWMR and (c) MWMR bandwidth allocation schemes.

Without the ability to share resources, this would inevitably greatly increase the number of MRRs required by the system and thereby increase the amount of thermal tuning power necessary to sustain tuning in the system. However, the nature of dynamic allocation allows a system architect to cut the number of wavelengths available to the overall system without reducing the maximum number of wavelengths available to a single switch. Therefore, if the system-level total number of wavelengths available is cut in half, the number of rings, and consequently, the thermal tuning power used to hold these rings at resonance, remains unchanged. If the total number of wavelengths is reduced further, the tuning power for the corresponding allocation scheme can be reduced even further below that of its static-allocation counterpart. However, tradeoffs between tuning power and throughput performance must then be considered.

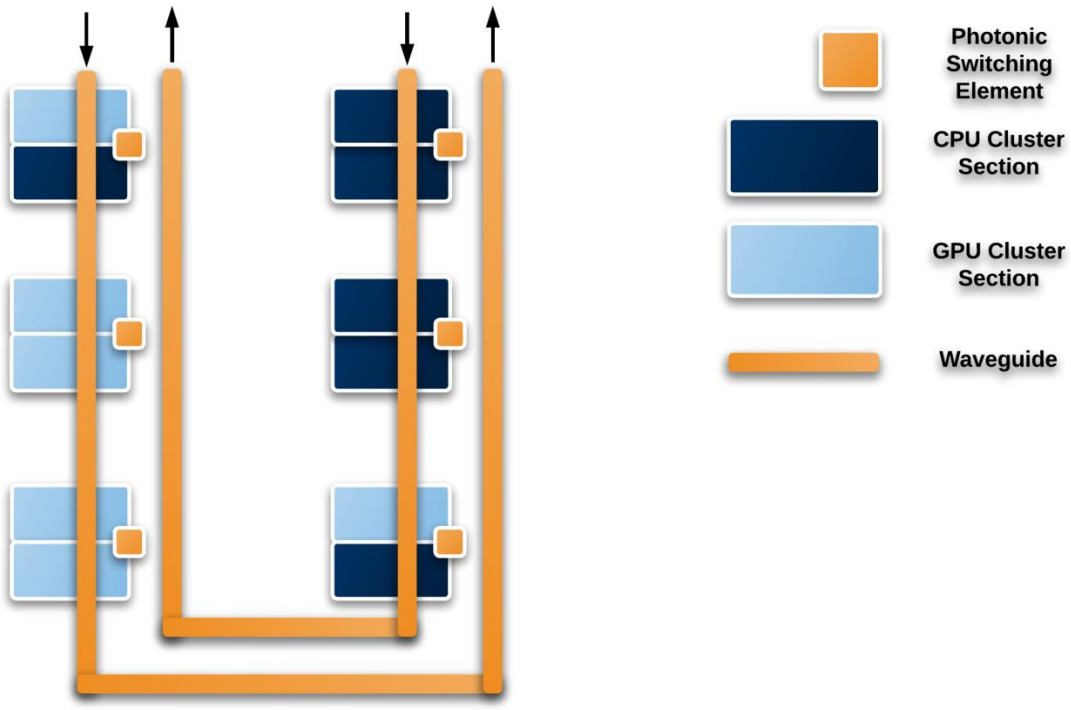

Figure 2: Layout of photonic components and cores. Adaptation of layout diagram in [31]

3.2. Heterogeneity

As the trends in computing towards heterogeneous processors accelerate, the fact that different processor types require different bandwidths becomes more apparent [24]. Consider a system with many multi-core tiles as shown in Figure 2. Some tiles may be like cluster 1 (shown in Figure 3(a)) and contain a CPU which is processing data from a PCIe 3.0 x8 device which results in a transmission bandwidth of 8 GB/s [39] (only considering transmission bandwidth) and a GPU which is connected to memory through a 29 GB/s data bus [23]. The total bandwidth demand of 37 Gigabytes per second could be managed by 8 photonic channels [43]. This would be classified as a maximum 8 or "M8" cluster because it would at most need 8 channels. Note, only the transmission bandwidth is considered for the PCIe 3.0 connection because the resources used for communication in the reverse direction would be allocated to the node with which the current cluster is

communicating as that would be the node that is transmitting in the reverse direction. Other tiles may be like cluster 2 (shown in Figure 3(b)) and contain two CPUs which are each working on separate PCIe 3.0 x4 channels and consequently require only 8 GB/s for the whole cluster for transmission. This would be classified as an M2 cluster because its traffic could be handled comfortably with 2 photonic channels.

Figure 3: (a) Cluster 1 and (b) cluster 2 with Photonic Switching Elements (PSEs) for network connectivity.

3.3. Data Channels and Arbitration

As opposed to transmitting allocation data over the main data channel which has a width of w bits, separate reservation channels are used. When compared to the allocation scheme of R-SWMR [8], the R-MWMR schemes incur a slight overhead in laser power. R-SWMR already features a reservation system which uses log N bits to transfer the address of an N-bit destination and log s bits to account for s different packet sizes. R-MWMR would include these as well as log B bits for B different possibilities of bandwidth requirements. Assuming 4 different possibilities, this would imply 2 bits of overhead compared to R-SWMR. The overheads based on [8] in area, $\log(NsB)/w$, static power, $(N\text{-}1)\log(NsB)/w$ and dynamic power, $\log(NsB)/(wt)$, considered with a few

example parameters (N = 8, s = 2, w = 256, t = 2 and B = 4), leads to overheads of 2.3%, 16%, and 8% respectively compared to a standard SWMR crossbar for the reservation channels. As the datapath width w increases, the relative overheads decrease. Moreover, k wavelengths for a credit stream are used as in [31] to communicate the buffer statuses of the nodes to one another where k is the radix of the crossbar.

However, with these increases comes the flexibility to decrease the total wavelengths which the system needs. Considering a potential 50% reduction in the number of supplied wavelengths for data waveguides which leads to a significant reduction in the pre-overhead laser power, the benefits outweigh the costs.

3.4. Medium Access Control

3.4.1 Allocation Mechanism Architecture Overview

Network-on-Chip designers are able to use various media for communication. At some point, a given node may require many system resources and then have a lower requirement at another moment. Moreover, the resources of the given medium can either be assigned solely to individual system nodes or shared by all nodes in the system. In the case for which nodes share the network's resources, either control of one or more of the network resources must be given to a node before usage to avoid collision or the collisions must be mitigated after the fact. In most cases, this resource allocation process either takes many cycles as with conventional token-based schemes or is vulnerable to collisions like CSMA [40]. Both of these problems become increasingly noticeable as the network scales and therefore, a reservation-based approach is used as in [8].

The goal is decentralized yet coherent allocation. In the same way that the result of 2 + 2 equals 4 on every computation unit, regardless of location, a proper decentralized

allocation mechanism must maintain consistency from node to node. Therefore, any allocator architecture will suffice if it can take in a set of resource requests and allocate non-overlapping sets of resources to each node and produce results that can be reproduced identically at each node. However, for a proof-of-concept study, a cellular architecture which allocates a system's available resources almost instantaneously is discussed below.

The principle of operation is drawn from other PNoC architectures [8] [31] which use reservation-based schemes to alert a receiver as to when to listen on certain channels. However, this work extends the principle of reservation to include the amount of resources allocated. The cellular nature of this architecture negates the need for a multi-cycle sorting algorithm in order to perform expeditious decentralized dynamic allocation.

3.4.2 Process Overview

At a very high level, the purpose of this allocation architecture is to enable each node individually through its respective allocator mechanism to determine which resources are free and allocate them to various network nodes. The network as a whole contains a set of resources which each have a corresponding resource number. The resource numbers of available resources are read from a register and then sorted in an order relative to their original resource numbers. After being sorted, allocation windows are applied to determine which resources are to be allocated to the network node performing the allocation and which resources are to be allocated to other nodes in the system. If nodes in the system receive the resources, the free resource register removes these resources from its records during their time of usage. If the node performing allocation is to hold on to the resources in a given window, the resource numbers are

stored locally and the allocation process repeats in the next cycle. For this work, the resources are taken to be the wavelengths available in a Photonic Network-on-Chip which employs WDM and the allocator architecture distributes wavelengths to routers (the network nodes).

The example shown in Figure 4 illustrates the allocation process in which three nodes are requesting resources. Only the allocation process at the second node is observed, however, this process occurs at each node in the system. The first and second nodes are requesting one resource each and the third node is requesting two resources. In Step 1, the free resources are assigned ordinal numbers (1^{st}, 2^{nd}, 3^{rd} etc.) relative to other free resources with lower earlier resource numbers. In Step 2, the allocation window corresponding to the allocator node, i.e. node 2, and the windows corresponding to other nodes are considered. In Step 3, the free resource register removes all allocated resources and the allocator node stores its newly-allocated resource references to a self-owned resource register.

3.4.3 Architecture Components

In this section, the key unit-level and cell-level design components are enumerated.

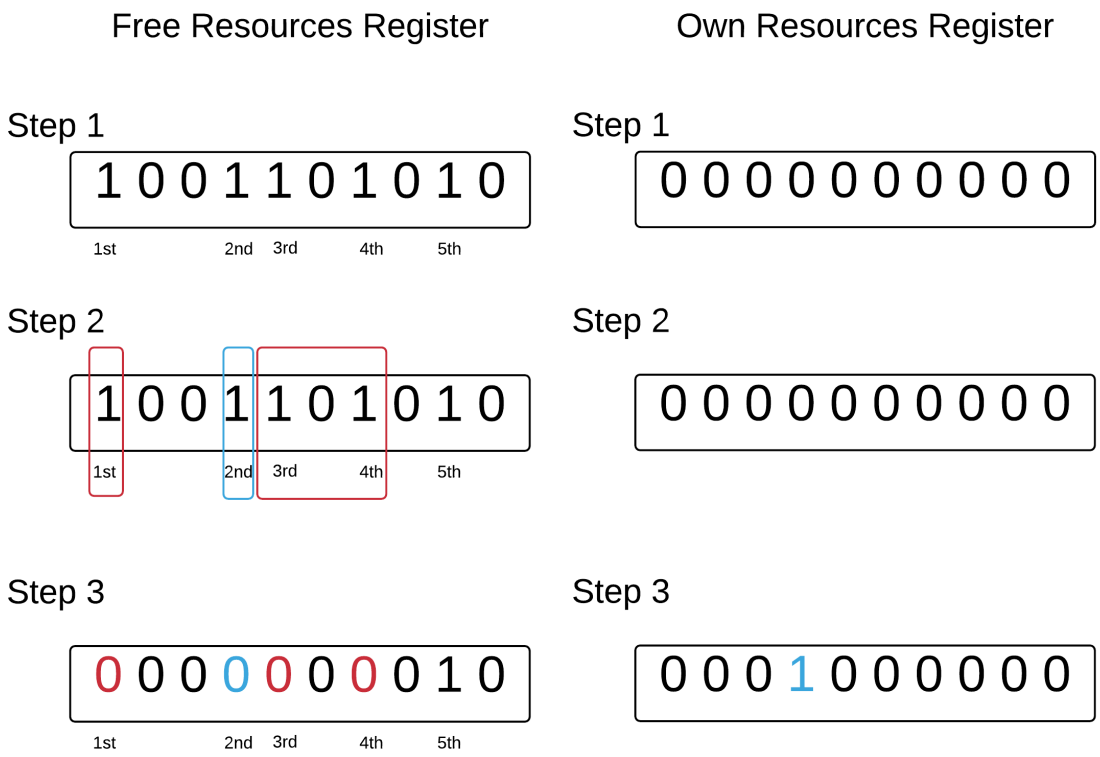

Figure 4: Allocation process example.

3.4.3.1 Medium Access Control Unit

The following components are integrated to create a medium access control unit as a whole as shown in Figure 5. The demands of the allocator node (the node performing allocation locally), the demands of other nodes and the state of the Free Resource Register are input into two subcomponents of the MAC unit which handle allocation to the allocator node itself (via the Local Alocation Module) as well as to the other nodes in the system (via the System Allocation Module).

3.4.3.2 Free Resource Register

The Free Resource Register (FRR) is responsible for maintaining a record of which resources are free and which are currently in use. The bit size of this register is equal to the number of resources in the system. The wavelengths owned by the allocator

at a given moment are stored in a similar register called the Own Resource Register (ORR).

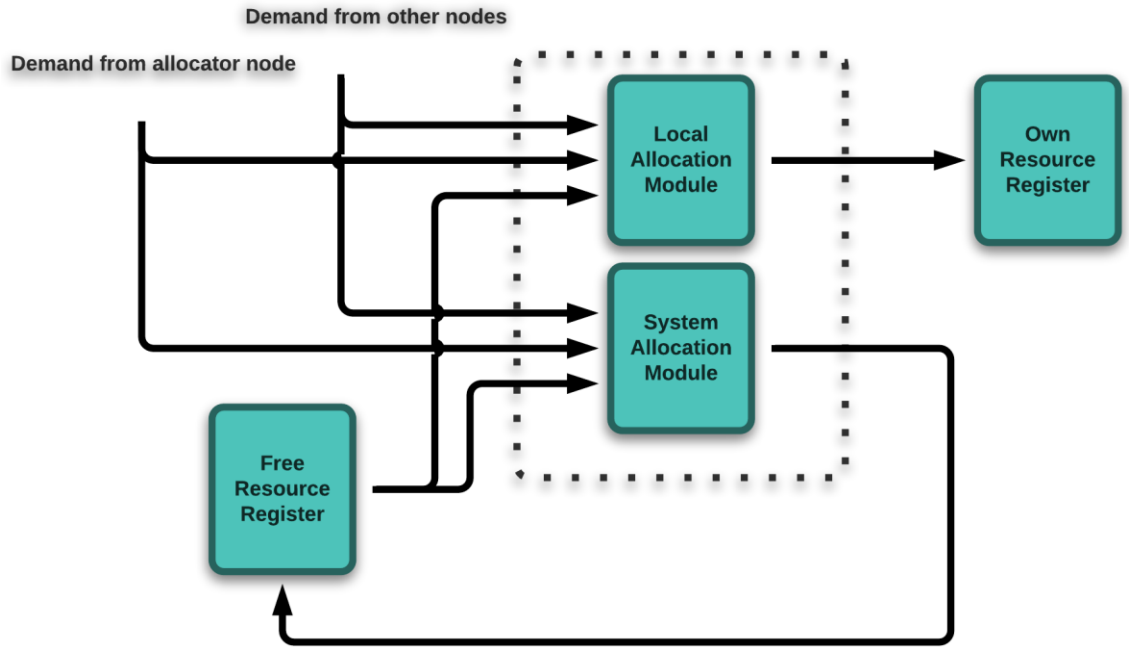

Figure 5: Overall medium access control unit

3.4.3.3 Allocation Modules

The Local Allocation Module (LAM) and the System Allocation Module (SAM) are responsible for reading information from the FRR about which resources are free and outputting the bit positions of resources which will be allocated to a given router. The allocation modules determine whether a resource number is within the window of resources to be allocated to the allocator or not. If so, the resource is then flagged to be written to ORR. If the resource number is within the window of resources to be allocated to any module in the system, the SAM clears it from the FRR.

3.4.3.4 Ordinal Marker Unit

The Ordinal Marker Unit exists within the allocation modules and determines the ordinal number of a given resource with respect to the other free resources. In this case, a Summation Unit accomplishes this purpose. The unit adds the bits of the FRR and outputs the cumulative sum after adding each bit from the most significant to the least as shown in Figure 6(a). The cumulative sum of the FRR up to the p-th bit is output to the p-th cell of the Allocation Modules (Local and System), where one cell corresponds to each bit of the FRR. If the ‘q’ number matches this sum, this indicates that there have been ‘q’ free resources encountered up to and including the current bit, where ‘q’ is the ordinal resource number . The p-th bit of the FRR register is then output. This ensures that of the resources for which the aforementioned condition is met, only one resource’s reference bit is triggered at the output. Note: Any unit which can determine the ordinal number of a resource can be used in place of

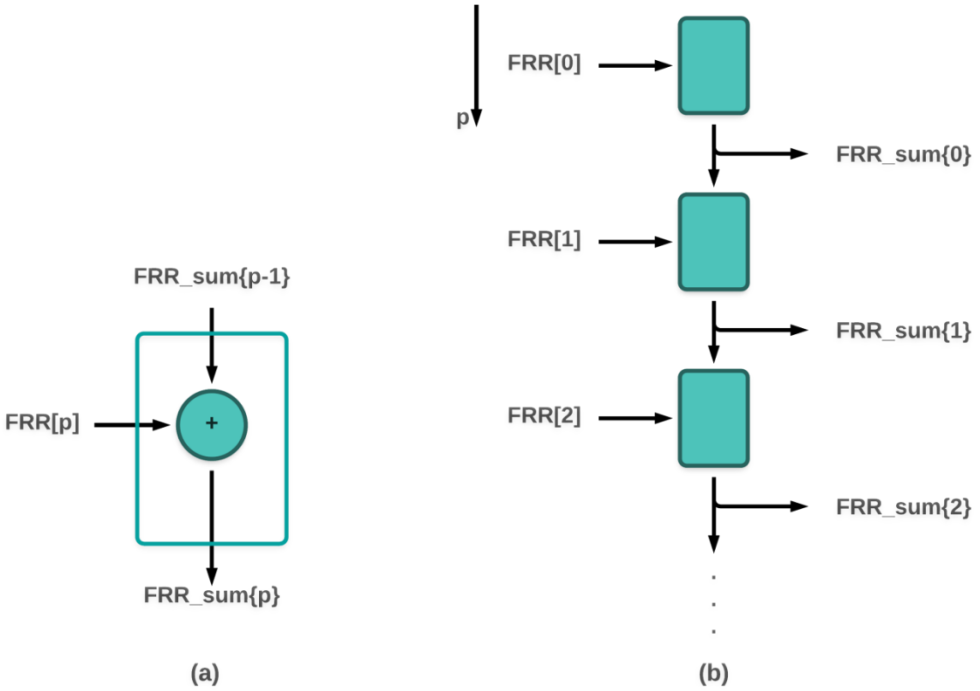

Figure 6: (a) Summation unit sub cell and (b) summation unit

Chapter 4 Dynamic Traffic Simulation

The system architectures under consideration are heterogeneous and therefore have varying bandwidth requirements from core to core. Moreover, systems have the potential to be not only heterogeneous at the network level, but also at the cluster level with each cluster containing cores of various bandwidth capabilities. The inter- and intra-cluster heterogeneity gives rise to bandwidth demands dependent upon which core within a given cluster is utilizing the network. At a system level, for the experiments carried out on a crossbar, the effect of inter-cluster heterogeneity is negligible as long as an overall traffic distribution is maintained. Therefore, for experiments, all clusters are allowed to make resource demands up to the maximum amount of resources possible (i.e. all cores are "M8"). For non-crossbar architectures, the hierarchy level of a node may affect allocation ability of the network.

Two notations are established for ease of reference to bandwidth requirements of clusters and network packets. λ_X describes traffic which requires X resources to match the speed of transmission from its source cluster. MY, or "Maximum Y," refers to a cluster which, at most, needs Y resources to sustain its most bandwidth-intensive communication. For example, if a cluster contains 4 cores which have maximum bandwidth needs of 1, 1, 2, and 4 resources, this cluster would be labeled as an M8 cluster because it needs at most 8 resources to support all of its transmissions in the event that they occur simultaneously. Even so, it can transmit slower traffic, e.g. λ_2 traffic. However, higher-bandwidth traffic cannot be directed to a lower-capability cluster, e.g. λ_8 traffic cannot be sent to an M4 cluster.

4.1. Medium Access Simulator

This section describes the requirements for creating a datalink-layer simulator for a network-on-chip environment which can handle dynamic bandwidth allocation. Attention is primarily on the components of such a simulator which go beyond those of a typical network-on-chip simulator. The components of the simulator as a standalone unit as well as an addition to a network-on-chip simulator are described as follows.

4.1.1 Demand

The demand component of the simulator is responsible for generating the resource requirements of all nodes at any given moment. When integrated into a network-on-chip simulator, this component can be included in the section of the NoC simulator which deals with the generation of header flits.

4.1.2 Allocation

The allocation component of the simulator is responsible for reading the resource requirements of nodes and allotting resources based on an allocation scheme. This is especially important with regard to handling resources allocation when a node has reached its per-node limit or if a node is requesting more resources than the system can offer at the time. For integration into a NoC simulator, the allocation component can be placed in the section of the NoC simulator which handles routing and transmission of header flits.

4.1.3 Resource Timing

The purpose of the resource timing component is to track when each flit passes between two nodes. After a full packet has been transmitted, the resource timer runs out

and alerts the system that the resources allocated to the transmitting node can be deallocated. For integration into a NoC simulator, the resource timing component can be placed in the sections of the NoC simulator which handle the transmission of header, body and tail flits. Moreover, in the event that sufficient resources are not available, the passing of flits is either restricted.

4.1.4 Deallocation

The purpose of the deallocation component is to release resources back into the system-wide pool once a node has completed its transmission. Integration into a NoC simulator involves placement of the deallocation component into the part of the NoC simulator which handles the passing of tail flits from one node to another.

4.2. Selection Rate

During the time of simulation, "traffic," or a stream of packets, comes from the nodes collectively as dictated by certain parameters. The generation of each packet in the simulator is based upon the *injection rate*, i.e. the probability of a node injecting a header flit into the network. Concurrent with injection, a corresponding resource requirement is attached to the packet based on a bandwidth *selection rate*. This selection rate, or the probability of a particular type of bandwidth requirement to be selected by a capable node in the system, determines what a packet bandwidth should be depending on a *demand profile* D which is specified at the beginning of the simulation.

$$\boldsymbol{D} = [\boldsymbol{d_1}\ \boldsymbol{d_2} \ldots\ \boldsymbol{d_i} \ldots\ \boldsymbol{d_N}] \qquad (1)$$

This profile contains elements with a subscript of X and dictates the percentage of the total simulation traffic which is to be λ_X traffic for all values of X defined in the simulator by the vector L. Note λ_X traffic is traffic which will require X wavelengths.

$$\boldsymbol{L = [1\ \ 2 \ldots\ l_i \ldots\ \ l_N]} \quad (2)$$

For the majority of experiments in this work, the following conditions hold:

$$\boldsymbol{D = [d_1\ d_2\ d_4\ d_8]} \quad (3)$$

$$L = [1\ \ 2\ 4\ 8] \quad (4)$$

Noteworthy, however, is that a node allocated 8 resources ideally finishes transmission eight times as quickly as a node allocated only one resource and so on for other amounts of resources. Therefore, not only must the probability of generation be considered for maintaining a demand profile, but the probability of a certain message type to leave the network as well. To address this, each message type probability must be scaled by the duration of time (under ideal conditions) for which a message remains in the network. The result is then scaled by the element sum to ensure that all corresponding probabilities add up to one. Because the likelihood of leaving the network is proportional to the amount of resources allocated, the following equation can be considered for generating the compensated demand profile elements $d_{i,comp}$ for all i. This can also be accounted for by multiplying the D vector by the ideal ejection rates and normalizing.

$$d_{i,comp} = \frac{d_i \cdot l_i}{\sum(D \cdot L)} \quad (5)$$

Table 1: Selection rate probability example (D = [0.2071 0.7000 0.0875 0.0054], i.e. a distribution with a mean demand of 2 and a demand variance of 0.75)

	M1	**M2**	**M4**	**M8**
$\boldsymbol{\lambda_1}$	1.0000	0.1289	0.1058	0.1036
$\boldsymbol{\lambda_2}$	0	0.8711	0.7153	0.7000
$\boldsymbol{\lambda_4}$	0	0	0.1788	0.1750
$\boldsymbol{\lambda_8}$	0	0	0	0.0214

Due to the inherently constant-bandwidth nature of most traffic generators in NoC simulators, a traffic generator supplement is added to an existing uniform traffic generator which works in conjunction with an in-house network-on-chip simulator.

4.3. Bandwidth Demand Generator

Given M_Y the event that an M_Y core is the source of a message in a given cycle and λ_X, when considered in probabilistic calculation, the event that λ_X traffic is selected at the generation of a flit in a given cycle, Bayes' theorem yields the following probability of a given core generating λ_X traffic with a given type of core.

$$P(\lambda_X | M_Y) = \frac{P(\lambda_X \cap M_Y)}{P(M_Y)} \quad (6)$$

Given that only one type of traffic can be generated at a time, the events can be considered disjoint and $P(M_Y)$, as per the total probability theorem is:

$$P(M_Y) = \sum_{X=1}^{Y} P(M_Y | \lambda_X) P(\lambda_X) = \sum_{X=1}^{Y} \frac{P(\lambda_X)}{N_X} \quad (7)$$

When the previous two equations are combined and Bayes' theorem is applied to the numerator of the first, the resulting equality is

$$P(\lambda_X | M_Y) = \frac{P(M_Y | \lambda_X) P(\lambda_X)}{P(\lambda_X)} \quad (8)$$

where $P(\lambda_X)$ is defined by the demand profile and $P(M_Y \mid \lambda_X)$ is equal for all N_X cores which are capable of supporting λ_X traffic and therefore equal to $1/N_X$. Relating the two equations yields the following probability of a core to generate λ_X traffic:

$$P(\lambda_X | M_Y) = \frac{\frac{P(\lambda_X)}{N_X}}{\sum_{X=1}^{Y} \frac{P(\lambda_X)}{N_X}} \; for\; Y \geq X;\; 0\; otherwise \quad (9)$$

Chapter 5 Allocation Performance Analysis

5.1. Dynamic and Static Allocation

For certain types of traffic, dynamic resource allocation performs extremely well. The noteworthy commonalities of these traffic distributions become apparent after simulating resource allocation performance when various allocation schemes are subjected to various traffic distributions.

A 64 node network is considered in conjunction with 5 allocation schemes and the average allocation of resources to each node is then observed. The first scheme (**Static4**) employs static bandwidth allocation. At the time of design, each node is given 4 resource units with which to communicate. Each node can, therefore, can use at most 4. The second scheme (**Static8**) is identical to Static4 except that 8 resource units are given to each node at the time of design. The static cases are chosen to be R-SWMR crossbars to ensure the same zero-load latency of a single-flit transmission, i.e. the only difference in the cases considered are the links themselves. The third allocation type is **Dynamic8-S4**. This scheme can allocate a variable number of resource units (up to 8) to each node. The total number of resources available to the whole system is equivalent to that of a Static4 scheme (i.e. 4 x 64 = 256). **Dynamic8-S2**, and **Dynamic8-S1** are similar with system-wide resources totals equivalent to those of a Static2 scheme and a Static1 scheme respectively. A Dynamic8-S8 scheme is implicitly considered in that its performance is identical to that of a Static8 system. From this, a system-wide mean μ_S can be defined as follows:

$$\mu_S = \frac{TOTAL\ SYSTEM\ RESOURCES}{NUMBER\ OF\ NODES} \quad (10)$$

Special attention is given to the Dynamic8-S4 allocation scheme in comparison with the Static8 and Static4 schemes which represent its effective upper and lower bounds in terms of allocation performance.

Initial results compare the performance of the allocation schemes in question with different traffic distribution vectors, e.g. 50% λ_1 traffic, 0% λ_2, 0% λ_4, and 50% λ_8. Upon inspection of the results in Figure 7, it can be observed that the results of some traffic distributions, more than others, tend toward the ideal result of the Dynamic8-S4 case matching the performance of Static8 which uses more power while, at the same time, outperforming and distinguishing itself from Static4 which uses less power although not necessarily less energy.

The problem, on a conceptual level can be broken down into two parts on the quest to finding the ideal traffic distribution for which a given dynamic allocation scheme is suitable. The first conceptual part is that Dynamic8-S4 can handle λ_8 traffic while Static4 cannot. Therefore, the more λ_8 traffic in the distribution, to a degree, the more Dynamic8-S4 will out-perform Static4. The second conceptual part is that due to the system-wide resource limit, the more λ_8 traffic in the system beyond a certain point, the more Static8 will outperform Dynamic8-S4. After testing the average resource allocation result of 100% λ_8 traffic, shown in Figure 7, the pattern can be observed that the patterns with higher variances and means closer to the system-wide average, e.g. 4 resources per node for Dynamic8-S4, have a better performance than their low-variance or high-mean counterparts. Experimentation is carried out on the allocation simulator to verify this. In Figure 7, 12.5 Gbps is assumed to be the capability of 1 resource as in [48].

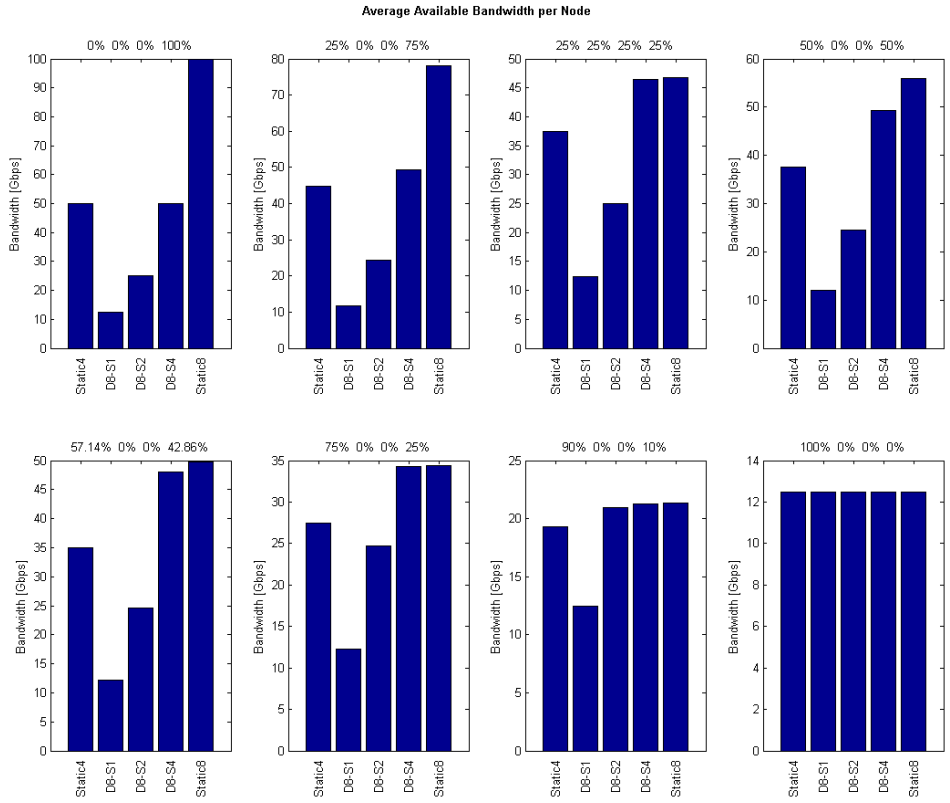

Figure 7: Simulation results for average allocation of resources to each node for Static4, Dynamic8-S1, Dynamic8-S2, Dynamic8-S4 and Static8 subject to various traffic distributions. Traffic percentages in each subtitle denote λ_1, λ_2, λ_4 and λ_8 traffic respectively. Destination selection is given by a bitwise complement (bitcomp) traffic pattern.

5.2. Proposal of the Ideal Traffic Distribution for Dynamic Allocation

Dynamic allocation schemes perform differently compared to static allocation schemes based on the resource demand distribution of nodes in the system at a given moment. As the results above point to higher variance and specific means leading to ideal performance, the extremes of these conditions are tested.

To analytically determine the traffic which has the highest variance σ_T^2, as characterized by the traffic variance equation (15), and a mean μ_T, as characterized by the traffic mean equation (14), equal to the system-wide mean, a Lagrangian optimization is carried out to

find a stationary point in the variance with respect to four traffic distribution elements which make up the traffic probability distribution D

$$D = [d_1\ d_2\ \ d_4\ d_8] \quad (11)$$

where d_x is the probability of a demand for x resources at a given moment. L denotes the x resources which can be allocated at a given time to a node with a probability of d_x.

$$L = [1\ \ 2\ \ 4\ \ 8] \quad (12)$$

From this, the expected amount of resources demanded on average equals the sum of each possible demand amount (i.e. each element of the vector *L*) times the probability of that particular demand amount (i.e. each element of the vector *D*).

$$E(L) = \sum_i l_i d_i \quad (13)$$

For a particular traffic pattern D, the expected demand μ_T can be obtained by taking the dot product of the two vectors:

$$\mu_T = D \cdot L \quad (14)$$

Likewise, the traffic variance σ_T^2 can be considered as:

$$\sigma_T^2 = D \cdot L^2 - (D \cdot L)^2 \quad (15)$$

Lagrangian maximization is carried out for the variance equation with the constraint that the mean of the traffic μ_T should be equal to the system-wide mean of resources available μ_S (e.g. a mean of 4 resources demanded per node for a Static4 system, a mean of 8 for Static8, etc.). The sum of the elements in the vector D is constrained to 1 given that the vector characterizes a probability distribution. The two center elements of D can be set to zero to further constrain the problem as doing this qualitatively increases the variance

even before optimization. The optimized values can then be verified (e.g. a variance of 12 for a traffic mean of 4, a variance of 10 for a traffic mean of 6, etc.)

5.3. Verification of the Ideal Traffic Distribution for Dynamic Allocation

As mentioned before, the ideal traffic pattern for dynamic bandwidth allocation minimizes the performance gap between Dynamic8-S4 and Static8 while maximizing the distance between the performance of Dynamic8-S4 and Static4. The following traffic gain function g(D) is adopted for comparison

$$\mathbf{g(D)} = \boldsymbol{\Delta_{S4} - \Delta_{S8}} \qquad \mathbf{(16)}$$

where Δ_{S4} is the average resource allocation per node of a Dynamic8-S4 system minus that of a Static4 system and Δ_{S8} is the average resource allocation per node of a Static8 system minus that of a Dynamic8-S4 system. This

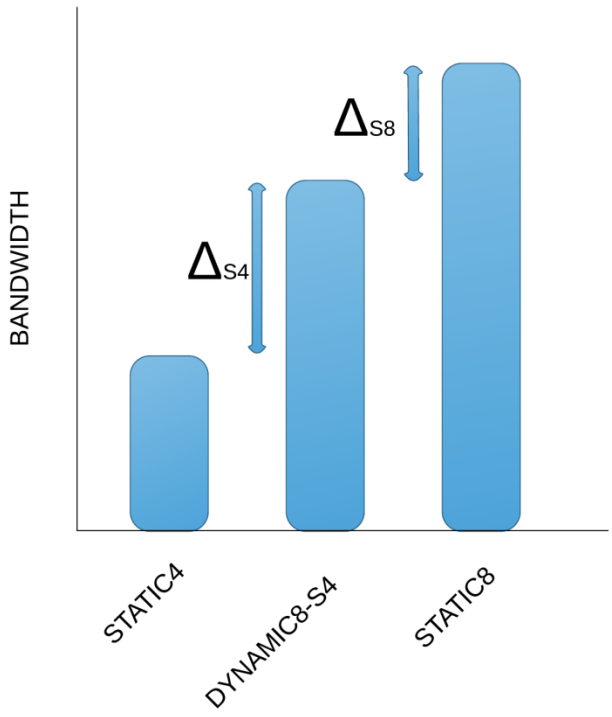

Figure 8: Gain Function components visualized on a graph showing the average amount of bandwidth allocated for Static4, Dynamic8-S4 and Static8 allocation in a given system.

Although the Lagrangian solution presented above maximizes the variance while maintaining a certain mean demand value for a given system, this result must be verified to determine if these two qualities actually lead to ideal system performance relative to Static8 and Static4. To do this, a simple simulator is developed to test allocation of resources at the level of medium access control, in isolation from other network characteristics which may create noise in the results due to network non-idealities.

In the simulator, each node demands resources in accordance with the probability distribution vector D. If the resources are available in the system, the required resources are taken from the system-wide pool of available resources and transferred to a specific node. This node retains these resources until it has finished transferring data to its destination at which point, it returns the resources to the system-wide pool. Given that the amount of resources given to a node is proportional to the bandwidth it has at its disposal, if a node has 8 resources, it finishes using its resources 8 times as fast and so on. Buffering limits and other network non-idealities have the potential to introduce noise into the system's performance results when making the initial establishment of a relationship between allocation and performance. Therefore, only the effect of allocation on performance is considered at first, but following the initial tests, the allocation mechanisms are implemented on a network-on-chip simulator to verify the results in the allocation simulator.

The simulation set up is pictured below in Figure 9. The initial set up allows the allocation simulator to run under similar conditions to the NoC simulator (i.e. same injection rate, same message length, simulation time, etc. and the average allocation numbers of each node are then assigned as the maximum bandwidth a node can support.

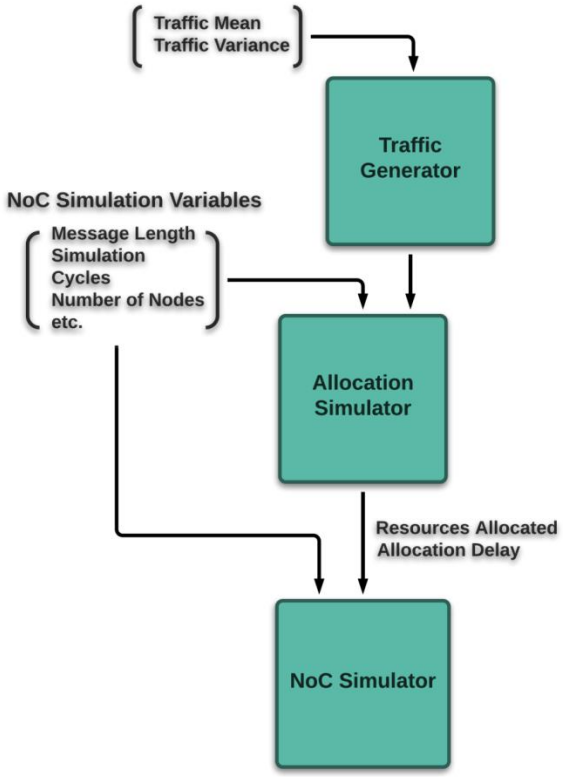

Figure 9: Simulation set up for examining dynamic bandwidth allocation schemes.

To verify the correlation between mean and variance with the performance of a dynamic system, traffic distributions of various means and variances must be generated. In order to do this in a deterministic manner, the mean and variance equations are set to specific values (μ_{set} and σ^2_{set} respectively) along with the constraint that the sum of the elements of D equals 1, as before in the Lagrangian optimization. For the final constraint, one variable is assigned a specific value. Newton's method of solving a system of non-linear equations is then implemented as follows:

$$\boldsymbol{D_1 = D_0 - \alpha\left[J\left(d_{1,0}, d_{2,0}, d_{4,0}, d_{8,0}\right)\right]^{-1} f\left(d_{1,0}, d_{2,0}, d_{4,0}, d_{8,0}\right)} \qquad \textbf{(17)}$$

Where α is a coefficient which dampens the effect of the Jacobian term, f is a vector made up of the set of four constraints mentioned above and shown below

$$\boldsymbol{f_1 = \sigma_T^2 - \sigma_{set}^2} \qquad \textbf{(18)}$$

$$f_2 = \mu_T - \mu_{set} \quad (19)$$

$$f_3 = \sum_x d_x - 1 \quad (20)$$

$$f_4 = d_2 - d_{2,constraint} \quad (21)$$

and J is the Jacobian matrix of f.

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial d_1} & \cdots & \frac{\partial f_1}{\partial d_8} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_4}{\partial d_1} & \cdots & \frac{\partial f_4}{\partial d_8} \end{bmatrix} \quad (22)$$

The $d_{2,constraint}$ term is assigned to values from 0 to 1 and for each value, until the convergence of all equations, Newton's method is executed for a fixed number of iterations. Moreover, a fully convergent distribution vector D_1 also satisfies the constraint that all of its elements are greater than or equal to zero. The higher the number of iterations of Newton's method, the more likely a point is to converge. Below are the points discovered in mean-variance space for integer values of mean and variance given 1, 2, 10, and 100 iterations of Newton's method; each point represents a traffic distribution vector.

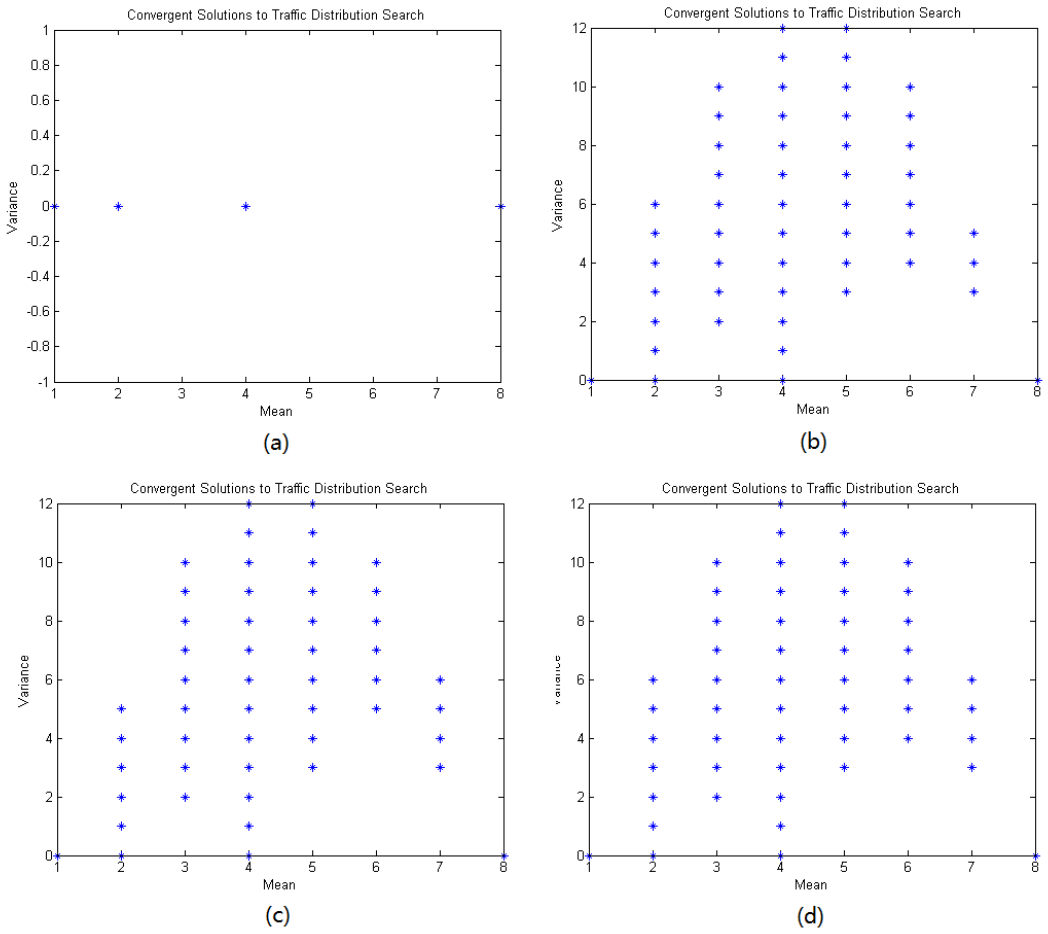

Figure 10: Potential traffic distributions for verification of the ideal traffic solution plotted in mean-variance space for (a) 1 iteration (b) 2 iterations (c) 10 iterations and (d) 100 iterations of the Newton's Method algorithm per value of $d_{2,constraint}$

After observation, it can be seen in Figure 10 that between 10 iterations and 100 iterations, there is little difference. More specifically, after 25 iterations, the majority of convergent points have been discovered by the algorithm. From this, a more complete set of points can be determined as shown in Figure 11 below.

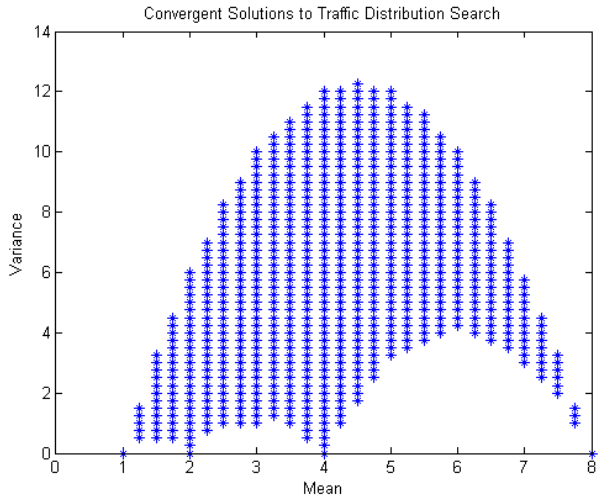

Figure 11: The traffic distributions used for verification of the ideal traffic solution plotted in mean-variance space.

Each point on the graph represents a traffic distribution with the corresponding mean μ_{set} and variance σ^2_{set} which can be simulated to verify that the solution determined previously is the ideal solution.

Figure 12 shows the values of the gain function g(D) to the fifth power as simulated for various values of mean and variance. (Dynamic8-S6) It is raised to the fifth power to accentuate vertical features of the graph while still maintaining the signs at all points.

As the plot tends toward the ideal point of maximum variance and a traffic-dependent mean which matches the system-wide mean, the gain function g(D) nears its maximum value. Dynamic8-S4 shows a similar convergence toward the maximum variance and the minimum difference between the system-wide resource mean and the mean of the traffic. For confirmation, the traffic distribution algorithm can be run with different initializations of the $d_{2,constraint}$ value and the same trend can be observed.

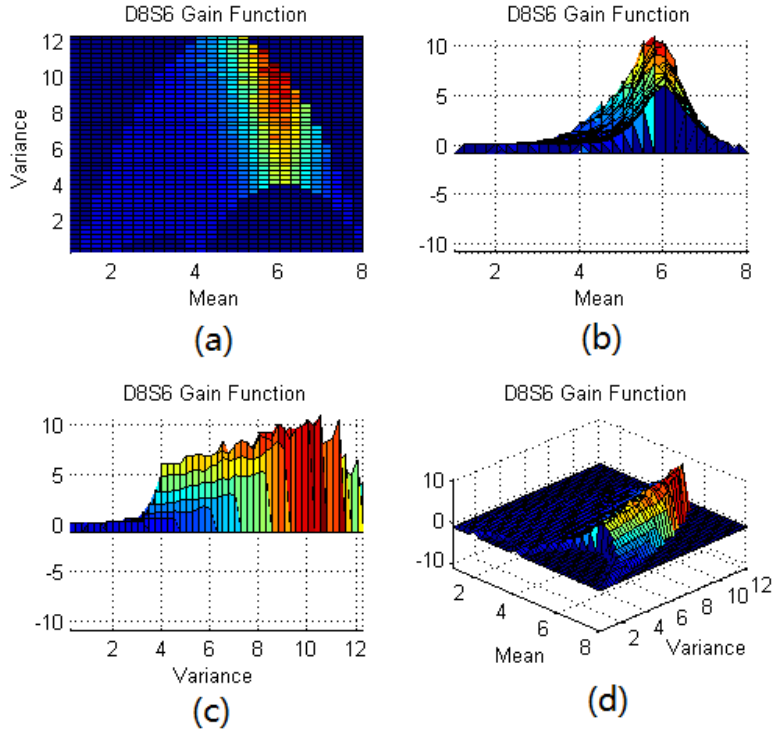

Figure 12: (a-d) The gain function for a Dynamic8-Static6 system under traffic of various means and variances from different angles.

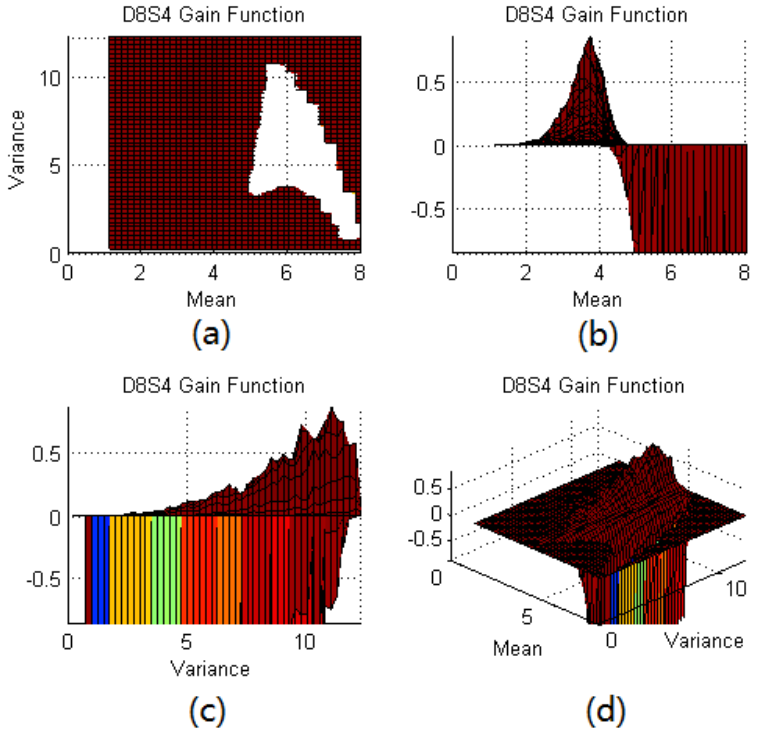

Figure 13: (a-d) The gain function for a Dynamic8-Static4 system under traffic of various means and variances from different angles.

5.4. Starvation

Beyond the amount of resources allocated to a node, the time it takes to allocate these resources must be considered as well. In Figure 14, the average allocation is shown after simulation of three schemes. **Flex8-Static4** (Flex8-S4) (a flexible allocation mechanism) is adapted from the architecture put forth in [31] in that it represents an allocation mechanism which shares resources among nodes but does not consider the individual resource needs of each node. This scheme always allocates the maximum (8 resources) to a node at the time of allocation and has a system-wide resource average of 4 resources per node. At first glance, its performance appears to match Dynamic8-S4, however two considerations must be factored in.

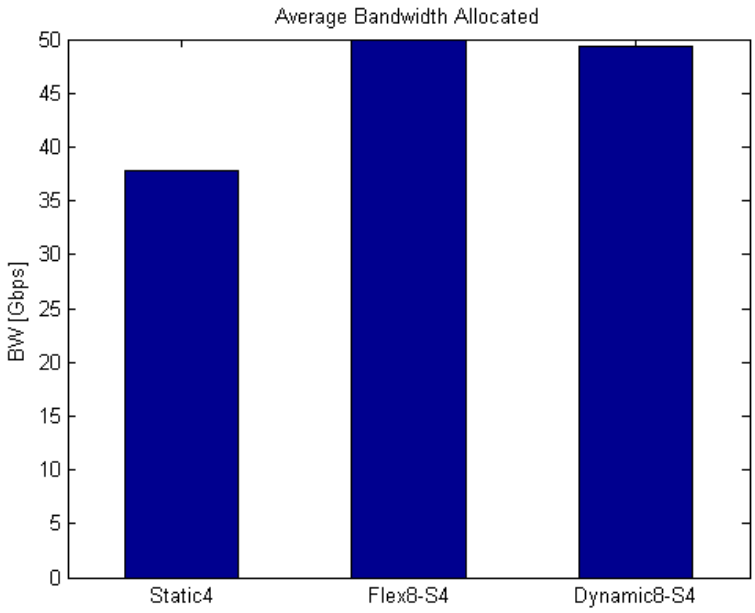

Figure 14: Average bandwidth allocation per node of various schemes following simulation of a traffic distribution with a mean of 4 and a variance of 12.

Firstly, the allocation mean of Flex8-S4 and other Flex8 variations always matches the system-wide mean of resources if the maximum per router is greater than or equal to the system-wide mean. The average amount of resources allocated in this case

can be deceiving because the transmitting node itself becomes a bottleneck which restricts data flow; no matter how many resources it is given above its bandwidth, the node will not out-do itself. Secondly because more resources are allocated where they are not needed, those same resources cannot be allocated where they are needed. As a result, resource starvation occurs. Figure 15 shows the average amount of cycles per packet for which resources were demanded but not allocated at various injection rates. Static4, by definition, has no starvation latency because of its nature. There is no allocation delay necessary for a resource to use its permanently allocated resources. Flex8-S4 on the other hand has an average starvation latency approaching 4 cycles while Dynamic8-S4 reaches only about the half a cycle at an injection rate of 1.0.

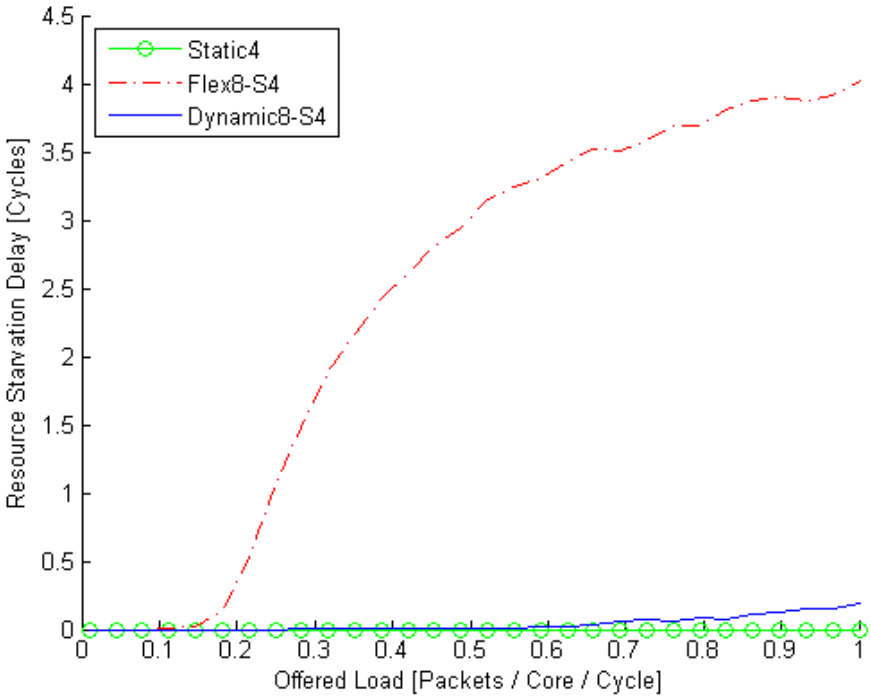

Figure 15: Average delay per packet due to starvation.

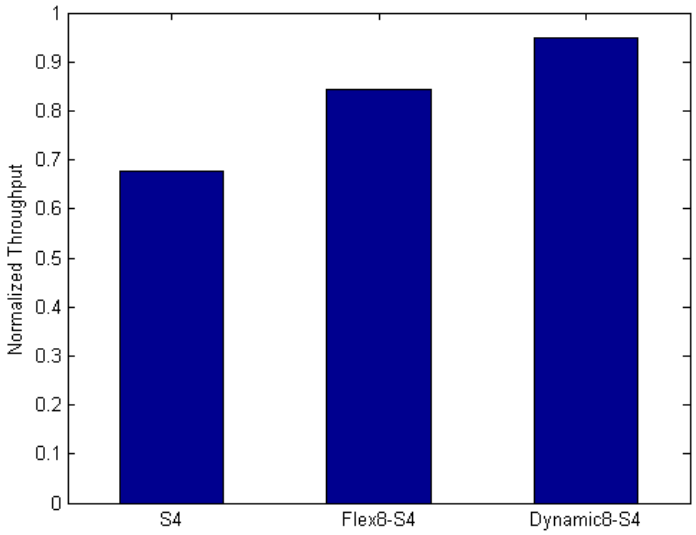

Figure 16: Throughput results when a bitcomp traffic distribution with a mean of 4 and a variance of 12 is applied to a system with various resource allocation schemes. From NoC simulator with performance normalized to that of a Static8 system

The Dynamic8-S4 scheme confirms its ability to handle varying traffic patterns in the NoC simulation throughput results shown in Figure 16. The throughput normalized to the performance of a Static8 system is displayed because this represents the maximum performance for Dynamic8-SX systems as well as Flex8-SX systems on a per-node basis. The low average allocation of the Static4 system leaves it in the lowest place with regard to throughput performance. The Flex8-S4 system falls behind the Dynamic8-S4 system because of allocation delay. The allocation delays of Flex8-SX schemes have an even bigger impact on the system performance as the mean of the traffic decreases because more resources are allocated where they are not needed, leaving other nodes starved. Dynamic8-S4 on the other hand is a fair allocation mechanism in that it allocates enough resources to where they are needed, but does not waste resources where they are not needed. Moreover, in comparison with Static4, Dynamic8-S4 allows nodes to have bandwidth above the system-wide mean without causing an increase at the system level.

5.5. Design for Injection Rate

The preceding results have for the most part considered an injection rate of 1.0 for the sake of pushing each allocation scheme to its limits. However, real-world applications often have far lower injection rates. One such example is the Map-Reduce benchmark [44] suite for multiprocessor computing. Various benchmarks are chosen from this suite for evaluation on various allocation schemes because of their variation in average injection rate. The simulation is based on traces from execution of the benchmarks on the gem5 [45] simulator. Figure 17 shows the average effective bandwidth per Node. The effective bandwidth is a combination of the latency determined by the bandwidth available to a node and the delay in allocating the bandwidth due to starvation. These benchmarks are evaluated on various allocation schemes. The injection rates are normalized to that of the most-frequently injecting node. Due to the low average injection rates (shown in Figure 18), a lower system-wide average resource amount μ_s can be provisioned while still maintaining high performance. Therefore, **Dynamic8-SX** and **Flex8-SX** schemes are given X (i.e. μ_S) values of 1, 2 and 4 while being compared to **Static2** and **Static8**.

Table 2: Description of schemes used in Map Reduce comparison

	Resources per Node	**Avg. Resources per Node**
Flex8-SX	Always (and only) 8	X
Dynamic8-SX	Up to 8	X
StaticX	Always (and only) X	X

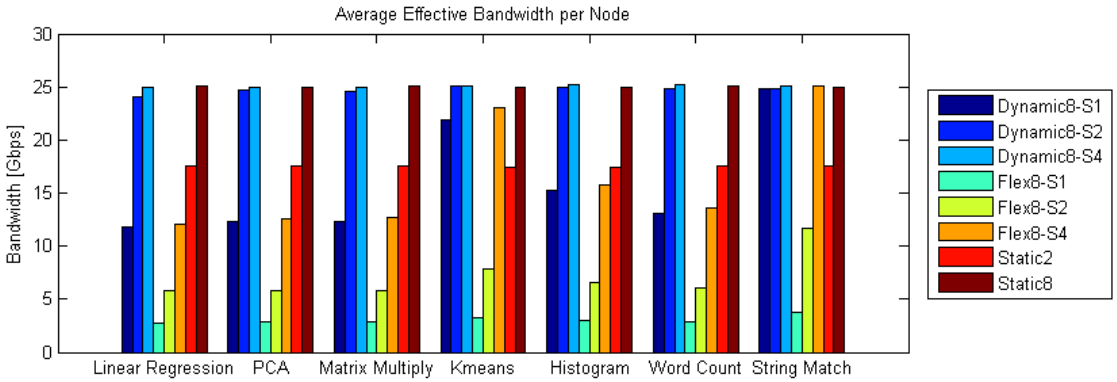

Figure 17: Average effective bandwidth per node for various benchmarks in the Map Reduce benchmark suite. Simulation Time: 10,000 Cycles. ($\mu_T = 2$ $\sigma_T^2 = 6$)

A direct correlation can be observed between the average injection of a particular benchmark and the performance of the Dynamic8-SX schemes relative to the Flex8-SX schemes. The higher the injection, the more likely the chance of starvation due to overprovisioning and therefore the better Dynamic8-SX performs when compared to Flex8-SX. The static allocation mechanisms, on the other hand, face the opposite problem; nodes are provisioned far below their requirements.

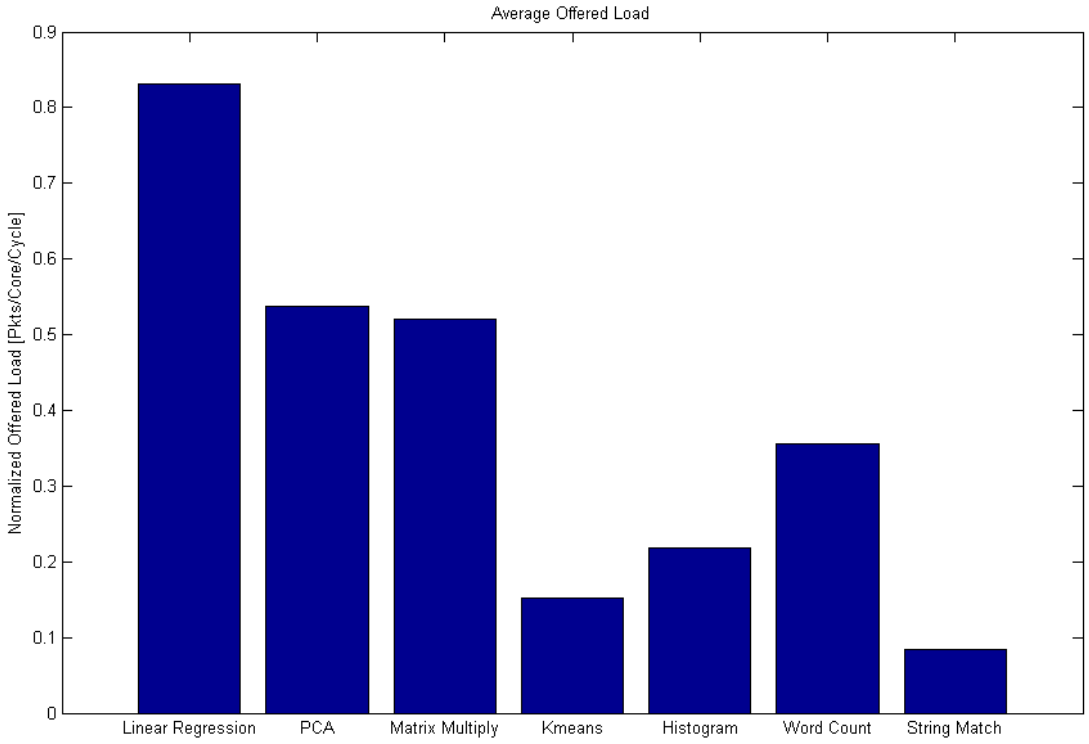

Figure 18: Average offered load per node for various benchmarks in the Map Reduce benchmark suite. Simulation Time: 10,000 Cycles. ($\mu_T = 2$ $\sigma_T^2 = 6$)

5.6. Energy

The energy consumed by the transmission of a given packet between links is given in Figure 19. The energy is calculated based on the power model in [31] and the necessary additions are made to account for the reservation channels used by the proposed allocation schemes. Although the Flex8-SX schemes do not require the marginal increase in power that Dynamic8-SX schemes need to accomplish reservation, the static power consumed while waiting for resources makes the Flex options less favorable.

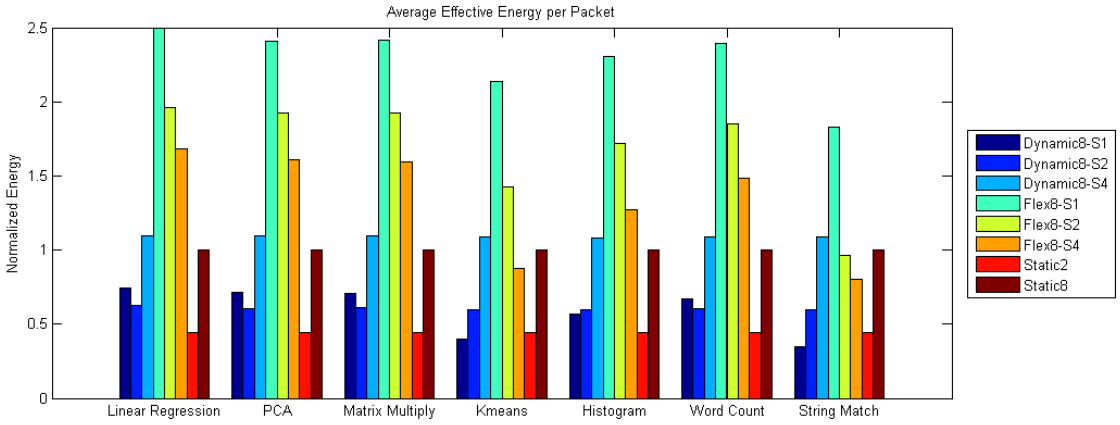

Figure 19: Energy of a packet between links for various benchmarks in the Map Reduce benchmark suite. Simulation Time: 10,000 Cycles. ($\mu_T = 2$ $\sigma_T^2 = 6$)

The Dynamic8-S2 case has approximately the same energy performance as the Static2 case while sustaining the performance of Static8. Furthermore, the Dynamic8-S1 scheme, due to the low injection rate of some benchmarks, is able to outdo the other schemes in terms of both energy and bandwidth.

Chapter 6 Future Work

Alongside the development of heterogeneous system simulators [35], dynamic bandwidth allocation simulation can be implemented as described in this paper as well as with variation in the cluster type, i.e. M1, M2, M4, etc., hence, introducing inter-cluster homogeneity. This inter-cluster homogeneity could then be observed in a hierarchical crossbar environment. In addition, many other papers which consider dynamic bandwidth allocation [46] [47] can be evaluated with a similar approach to the bandwidth needs of heterogeneous systems. The reservation-based scheme proposed in this work can allocate mixed resources, i.e. CDMA sub-channels on a photonic channel; this could be further explored to make the system even more dynamic.

.

Chapter 7 Conclusion

For Photonic Network-on-Chip systems, many options are available as far as resource allocation is concerned. However, after considering the low average resource allocation exhibited by a Static4 scheme, the high allocation latency a Flex8-S4 scheme experiences (which is aggravated by low-mean, high injection traffic distributions), and the large amount of energy a Static8 scheme consumes, the Dynamic8 schemes prove to be the most versatile and efficient under various kinds of traffic. Dynamic allocation schemes in general have proven to perform especially well when (1) the system-wide mean resources per node μ_S for the allocation scheme (e.g. 4 in the case of Dynamic8-S4) most closely matches a given traffic distribution's mean resource demand μ_T and (2) the variance of the given distribution is maximized. To minimize overhead due to thermal tuning of MRRs, when using a dynamic system, the average resources per node μ_S should be equal to or less than half of the per-router maximum of the scheme it is replacing (e.g. $\mu_S \leq 8/2 = 4$ for replacements of Static8). As far as allocation logic is concerned, if a distributed allocation system is implemented, the latency overhead due to allocation can be effectively hidden by the existing latencies of reservation in an R-SWMR statically-provisioned. Overall, dynamic bandwidth allocation is a viable solution to the needs of the heterogeneous multicore computing era.

Bibliography

1. P. Dziurzanski, T. Maka, G. Ulacha and R. Stasinski, "A lossless compression system realization utilizing phit-serial network-on-chip paradigm," in *15th European Signal Processing Conference*, Poland, 2007

2. A. Shacham, K. Bergman, and L. P. Carloni, "On the design of a photonic Network-on-Chip," in *Proc. NOCS*, 2007, pp. 53–64

3. R. Ho, K. W. Mai, M. A. Horowitz, "The Future of Wires," in *Proceedings of the IEEE, Vol. 89, No. 4*, April 2001

4. W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon, "Demystifying 3D ICs: The Pros and Cons of Going Vertical," in *IEEE Design & Test of Computers*, 2005, pp. 498–510

5. A. Ganguly, K. Chang, S. Deb, P. P. Pande, B. Belzer, C. Teuscher, "Scalable Hybrid Wireless Network-on-Chip Architectures for Multi-Core Systems," in *IEEE Transactions on Computers, Vol. 60, No. 10*, October 2011

6. K. Chang, S. Deb, A A. Ganguly, X. Yu, S. P. Sah, P. P. Pande, B. Belzer, D. HEO, "Performance Evaluation and Design Trade-Offs for Wireless Network-on-Chip Architectures," in *ACM Journal on Emerging Technologies in Computing Systems, Vol. 8, No. 3, Article 23*, August 2012

7. D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. P. Jouppi, M. Fiorentino, A. Davis, N. Binkert, R. G. Beausoleil, and J. Ho Ahn "Corona: System Implications of Emerging Nanophotonic Technology" in *Proceedings of the 35th Annual International Symposium on Computer Architecture (ISCA '08)*, Washington, DC, USA, 2008, pp. 153-164

8. Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang A. Choudhary. "Firefly: illuminating future network-on-chip with nanophotonics in *Proceedings of the 36th annual international symposium on Computer architecture (ISCA '09)*, New York, NY, USA, 2009, pp. 429-440

9. J. T. Kim and S. Park, "The Design and Analysis of Monolithic Integration of CMOS-Compatible Plasmonic Waveguides for On-Chip Electronic–Photonic Integrated Circuits," in *Journal of Lightwave Technology, Vol. 31, No. 18*, September 2013

10. Y. J. Hsu, B. H. Cheng, Y. Lai, D. P. Tsai, "Classical Analog of Electromagnetically Induced Transparency in the Visible Range With Ultra-Compact Plasmonic Micro-Ring Resonators," in *IEEE JOURNAL OF SELECTED TOPICS IN QUANTUM ELECTRONICS, VOL. 21, NO. 4*, July/August 2015

11. Z. Lu, W. Zhao, K. Shi, "Ultracompact Electroabsorption Modulators Based on Tunable Epsilon-Near-Zero-Slot Waveguides," in *IEEE Photonics Journal Volume 4, Number 3*, June 2012, pp. 735-740

12. V. J. Sorger, N. D. Lanzillotti-Kimura, R. M. Ma, X. Zhang, "Ultra-compact silicon nanophotonic modulator with broadband response," 2012

13. W. Bogaerts, P. D. Heyn , T Van Vaerenbergh , K De Vos, S. K. Selvaraja, T. Claes, et al. "Silicon microring resonators," *Laser & Photonics Reviews*, 2012, pp. 47–73

14. D. Siracusa, V. Linzalata, G. Maier, A. Pattavina, Y. Ye, M. Chen, "Hybrid Architecture for Optical Interconnection based on Micro Ring Resonators," *IEEE Globecom*, 2011

15. K. Iga, "Vertical-Cavity Surface-Emitting Laser (VCSEL)," *in Proceedings of the IEEE Vol. 101, No. 10*, October 2013

16. Y. Li, D. Vermeulen, Y. De Koninck, G. Yurtsever, G. Roelkens, R. Baets, "Fiber couplers for silicon-on-insulator photonic IC's with optimized on-chip return loss," in *OFC/NFOEC Technical Digest*, 2012

17. Z. Li, A. Qouneh, M. Joshi, W. Zhang, X. Fu, T. Li, "Aurora: A Cross-Layer Solution for Thermally Resilient Photonic Network-on-Chip," in *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, 2014

18. C. Condrat, P. Kalla, S. Blair, "Thermal-aware Synthesis of Integrated Photonic Ring Resonators," in *ICCAD '14 Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design*, 2014, pp. 557-564

19. K. Padmaraju, D. F. Logan, T. Shiraishi, Member, J. J. Ackert, A. P. Knights, and K. Bergman, "Wavelength Locking and Thermally Stabilizing Microring Resonators Using Dithering Signals," in *JOURNAL OF LIGHTWAVE TECHNOLOGY, VOL. 32, NO. 3*, February 2014

20. X. Tan, M. Yang, L. Zhang, Y. Jiang, J. Yang, "A Generic Optical Router Design for Photonic Network-on-Chips," in *JOURNAL OF LIGHTWAVE TECHNOLOGY, VOL. 30, NO. 3*, February 2012

21. R. Ji, L. Yang, L. Zhang, Y. Tian, J. Ding, H. Chen, Y. Lu, P. Zhou, and W. Zhu, "Five-Port Optical Router for Photonic Networks-on-Chip," in *OPTICS EXPRESS, Vol. 19, No. 21,* October 2011

22. F. Y. Liu, D. Patil, J. Lexau, P. Amberg, M. Dayringer, J. Gainsley, H. F. Moghadam, X. Zheng, J. E. Cunningham, A. V. Krishnamoorthy, E. Alon, R. Ho, "10-Gbps, 5.3-mW Optical Transmitter and Receiver Circuits in 40-nm CMOS," in *IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 47, NO. 9,* September 2012

23. A. Branover， D. Foley， M. Steinman, "AMD FUSION APU: LLANO," in *IEEE Micro Vol. 32, No. 2,* 2012, pp. 28-37

24. D. Bouvier, B. Cohen, W. Fry, S. Godey, "Kabini: An AMD Accelerated Processing Unit System on A Chip," in *IEEE Micro Vol. 34, No. 2,* 2014, pp. 22-33

25. M. Shevtsov, "OpenCL: The Advantages of Heterogeneous Approach" (via Intel Corporation)

26. A. K. Mishra, N. Vijaykrishnan, C. R. Das, "A Case for Heterogeneous On-Chip Interconnects for CMPs," in *2011 38th Annual International Symposium on Computer Architecture (ISCA),* 2011, pp. 389-399

27. E. S. Chung, P. A. Milder, J. C. Hoe, K. Mai, "Single-Chip Heterogeneous Computing: Does the Future Include Custom Logic, FPGAs, and GPGPUs?," *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO),* 2010, pp. 225-236

28. H. Calandra, R. Dolbeau, P. Fortin, J. Lamotte, I. Said, "Evaluation of successive CPUs/APUs/GPUs based on an OpenCL finite difference stencil," in *2013 21st Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP),* 2013, pp. 405-409

29. X. Wu, Y. Ye, W. Zhang, W. Liu, M. Nikdast, X. Wang, J. Xu, "UNION: A Unified Inter/Intra-Chip Optical Network for Chip Multiprocessors," in *2010 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH),* 2010, pp. 35-40

30. A. Joshi, C. Batten, Y. Kwon, S. Beamer, I. Shamim, K. Asanovic, V. Stojanovic, "Silicon-Photonic Clos Networks for Global On-Chip Communication," in *NoCS 2009. 3rd ACM/IEEE International Symposium on Networks-on-Chip,* May 2009, pp. 124-133

31. Y. Pan, J. Kim, G. Memik, “FlexiShare: Channel Sharing for an Energy-Efficient Nanophotonic Crossbar,” in *IEEE 16th International Symposium on High Performance Computer Architecture (HPCA)*, 2010, pp. 1-12

32. C. Li, M. Browning, P. V. Gratz, S. Palermo, “LumiNOC: A Power-Efficient, High-Performance, Photonic Network-on-Chip for Future Parallel Architectures,” in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 33, No. 6*, 2014, pp. 826-838

33. C. Sun, C. O. Chen, G. Kurian, L. Wei, J. Miller, A. Agarwal, L. Peh and V. Stojanovic, “DSENT – A Tool Connecting Emerging Photonics with Electronics for Opto-Electronic Networks-on-Chip Modeling,” in *Sixth IEEE/ACM International Symposium on Networks on Chip (NoCS)*, 2012, 201-210

34. J. Chan, G. Hendry, A. Biberman, K. Bergman, L. P. Carloni, “PhoenixSim: A Simulator for Physical-Layer Analysis of Chip-Scale Photonic Interconnection Networks,” in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2010, pp. 691-696

35. J. Power, J. Hestness, M. S. Orr, M. D. Hill, D. A. Wood, “gem5-gpu: A Heterogeneous CPU-GPU Simulator,” in *Computer Architecture Letters*, 2015, pp. 34-36

36. P. Dong, S. Liao, D. Feng, H. Liang, R. Shafiiha, N. Feng, G. Li, X. Zheng, A. V. Krishnamoorthy, M. Asghari, “"Tunable High Speed Silicon Microring Modulator," *Conference on Lasers and Electro-Optics (CLEO) and Quantum Electronics and Laser Science Conference (QELS)*, May 2010, pp.1,2

37. Y. Yang, P. Xiang, M. Mantor, H. Zhou, “CPU-Assisted GPGPU on Fused CPU-GPU Architectures,” in *IEEE 18th International Symposium on High Performance Computer Architecture (HPCA)*, 2012, pp. 1-12

38. Y. Ma1, Z. Xuan, Y. Liu, R. Ding, Y. Li, A. E. Lim, G. Lo, T. Baehr-Jones, M. Hochberg, “Silicon Microring Based Modulator and Filter for High Speed Transmitters at 1310 nm,” in *2014 IEEE Optical Interconnects Conference*, 2014, pp. 23-24

39. PCIE Express® Base Specification Revision 3.0, November 2010, pp. 40

40. Y. Chen, H. Wang, “Ordered CSMA: A Collision-Free MAC Protocol for Underwater Acoustic Networks,” in *OCEANS 2007*, pp. 1-6

41. P. Abad, P. Prieto, L. Menezo, A. Colaso, V. Puente, J. Gregorio, “TOPAZ: An Open-Source Interconnection Network Simulator for Chip Multiprocessors and Supercomputers,” in *2012 Sixth IEEE/ACM International Symposium on Networks on Chip (NoCS)*, May 2012, pp. 99-106

42. N. Jiang, D. Becker, G. Michelogiannakis, J. Balfour, B. Towles, J. Kim, W. Dally, “A Detailed and Flexible Cycle-Accurate Network-on-Chip Simulator,” in *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2013, pp 86-96

43. L. Xu, W. Zhang, Q. Li, J. Chan, H.L.R. Lira, M. Lipson, K. Bergman, “40-Gb/s DPSK Data Transmission Through a Silicon Microring Switch,” *IEEE Photonics Technology Letters, Vol. 24, No. 6*, March 2012, pp. 473-475

44. C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski, C. Kozyrakis, "Evaluating MapReduce for Multi-core and Multiprocessor Systems," in *IEEE 13th International Symposium on High Performance Computer Architecture, 2007. HPCA 2007*, February 2007, pp. 13-24

45. N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M.D. Hill, D. A. Wood, "The gem5 Simulator," in *ACM SIGARCH Computer Architecture News, Vol. 39, No. 2*, August 2011

46. Y. Xu, J. Yang, R. Melhem, “Channel Borrowing: An Energy-Efficient Nanophotonic Crossbar Architecture with Light-Weight Arbitration,” in *ICS '12 Proceedings of the 26th ACM international conference on Supercomputing*, New York, NY, USA, 2012, pp. 133-142

47. A. Zulfiqar, P. Koka, H. Schwetman, M. Lipasti, X. Zheng, A. Krishnamoorthy, “Wavelength Stealing: An Opportunistic Approach to Channel Sharing in Multi-chip Photonic Interconnects,” in *MICRO-46 Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*, New York, NY, USA, pp. 222-233

48. A. Shah, N. Mansoor, B. Johnstone, A. Ganguly, S. Lopez Alarcon, “Heterogeneous Photonic Network-on-Chip with Dynamic Bandwidth Allocation,” in 2014 27th IEEE International System-on-Chip Conference (SOCC), Las Vegas, NV, USA, September 2014, pp. 249-254